Exploring Geometry-aware Contrast and Clustering Harmonization for Self-supervised 3D Object Detection

Hanxue Liang[1]*, Chenhan Jiang[2]*, Dapeng Feng[3], Xin Chen[4], Hang Xu[2],
Xiaodan Liang[3]†, Wei Zhang[2], Zhenguo Li[2], Luc Van Gool[1],
[1]ETH Zurich [2]Huawei Noah's Ark Lab [3]Sun Yat-Sen University [4]The University of Hong Kong

Abstract

Current 3D object detection paradigms highly rely on extensive annotation efforts, which makes them not practical in many real-world industrial applications. Inspired by that a human driver can keep accumulating experiences from self-exploring the roads without any tutor's guidance, we first step forwards to explore a simple yet effective self-supervised learning framework tailored for LiDAR-based 3D object detection. Although the self-supervised pipeline has achieved great success in 2D domain, the characteristic challenges (e.g., complex geometry structure and various 3D object views) encountered in the 3D domain hinder the direct adoption of existing techniques that often contrast the 2D augmented data or cluster single-view features. Here we present a novel self-supervised 3D Object detection framework that seamlessly integrates the geometry-aware contrast and clustering harmonization to lift the unsupervised 3D representation learning, named GCC-3D. First, GCC-3D introduces a Geometric-Aware Contrastive objective to learn spatial-sensitive local structure representation. This objective enforces the spatially close voxels to have high feature similarity. Second, a Pseudo-Instance Clustering harmonization mechanism is proposed to encourage that different views of pseudo-instances should have consistent similarities to clustering prototype centers. This module endows our model semantic discriminative capacity. Extensive experiments demonstrate our GCC-3D achieves significant performance improvement on data-efficient 3D object detection benchmarks (nuScenes and Waymo). Moreover, our GCC-3D framework can achieve state-of-the art performances on all popular 3D object detection benchmarks.

1. Introduction

LiDAR-based 3D object detection has been a long-standing task in visual perceptions systems for autonomous driving, attracting increasing industry and research attention recently due to its great advantage of high 3D localization precision and complementary to the 2D perception [43, 2, 29, 47, 33]. Different from the 2D detection problem, 3D object detectors transform sparse and unorganized point clouds into the structured 3D bounding box representations, including shape, orientation and semantic class. Almost all recent 3D object detectors are built upon fully supervised frameworks, while obtaining such large-scale and precise annotations for numerous instances in diverse self-driving scenarios is labor-intensive and time-consuming, *e.g.*, it takes hundreds of hours to annotate just one hour of driving scene data [31]. This hinders the model improvement and deployment over ever-changing self-driving environments for LiDAR-based 3D object detection. Thus, a desirable self/unsupervised 3d object detection framework that can effortlessly lift the 3d representation learning purely using raw data is highly demanded but rarely explored.

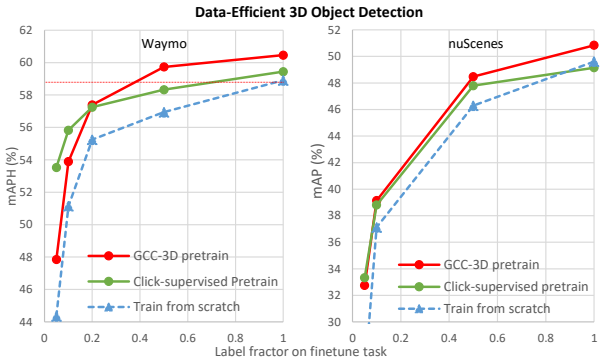

Figure 1. We finetune CenterPoint-pp detector from scratch, with GCC-3D pretrain or with Click-supervised pretrain and report performance on Waymo and nuScenes dataset. Our GCC-3D model show consistent significant improvements over the scratch model and learn more robust feature than Click-supervised pre-train

Nonetheless, in the area of 2D image recognition [30, 20, 14] and natural language understanding [12], self-supervised pre-training over unlabeled data has yielded a significant performance boosting in downstream tasks when the labeled data is scarce. Thus, it is interesting to ask a

*Both authors contributed equally to this work.
†Corresponding Author: xdliang328@gmail.com

question: *Does there also exist an effective self-supervised pre-training algorithm that can significantly alleviate the heavy annotation burden in 3D object detection by fully exploiting abundant unlabeled point cloud data?* Existing works mostly focus on low-level task [50, 10, 16] (*e.g.*, registration) and single object [17, 9, 1, 21] (like reconstruction, classification and part segmentation). Recently, PointContrast [44] demonstrated that unsupervised pretraining can boost the performance on indoor scene understanding tasks. However, several limitations of PointContrast hinder its direct adoption into LiDAR-based 3D detection: 1) Static Partial Views: Multiple partial views [53] setup is claimed to be a crucial component of [44], requiring the object/scene to be static. This is usually not available in outdoor scenes of autonomous driving scenarios. 2) Inconsistent Contrast: It assigns hard labels to matched and unmatched pairs, which is contradictory to the fact that the randomly sampled unmatched pairs can have very similar structure; 3) Lack of Semantic Information: Semantic representation is important for high-level scene understanding tasks like 3D object detection. This kind of representation is not modeled during pre-training.

To advance the research on LiDAR-based 3D object detection into an unsupervised/self-supervised era and resolve the above-mentioned issues in designing a proper self-supervised scheme, we present a novel self-supervised 3D detection framework that seamlessly integrates the geometric-sensitive and semantic-consistent representations, named GCC-3D. Our framework is the first one focusing on autonomous driving scenario without static partial views setup [53]. **Firstly**, to alleviate the inconsistent contrast problem, GCC-3D exploits an important property of 3D data: two spatially close voxels in 3D space are very likely to have similar local geometric structures or belong to the same object. We inject this prior to our learning objective and use the geometric distance between voxels to guide feature similarity during contrastive feature learning. This geometric-aware contrastive objective can help learn local structural features of point clouds properly. The voxel-level features with geometric information will be aggregated as the embedding of pseudo instances, which are obtained from the sequential information in datasets. **Secondly**, we endow our model semantic property by defining a clustering harmonization phase. During training, we assign labels to each instance embedding by using K-means clustering following [41]. However, the commonly used hard labeling strategy [41] violates that some prototypes can be similar or represent the same semantic classes, and neglects the heterogeneous similarities between embeddings of pseudo-instances and is prone to "class collision" [3] problem. To alleviate this problem, we introduce a new harmonization term that encourages different pseudo-instances views to have consistent similarities with clustering prototype centers. This term is easily injected into the current self-clustering framework. By integrating the geometry-aware contrast and pseudo-instance clustering harmonization, our GCC-3D can capture both local structure and context semantic information, which can improve our model's localization and classification capacity.

To better validate the self-supervised capability of current models in Lidar-based 3D object detection, we conduct extensive experiments in popular 3D object detection benchmarks (Waymo [39], nuScenes [5]) with limited supervised data, called data-efficient benchmarks. The methods are required to first pre-train only on the unlabeled data and then fine-tune it using limited labeled data to reduce the annotation effort. Our unsupervised framework GCC-3D can achieve consistent significant improvement over random initialized models on the data-efficient benchmarks. Notably, our pre-trained CenterPoint-voxel model achieves 67.29% mAP on Waymo (with 20% labeled data) and 57.3% mAP on nuScenes, a separate 4.1% and 1.95% relative improvement over previous state-of-the-art method [49]. After transferring our pre-trained model on Waymo to KITTI [18], we witness a 2.1% relative improvement over KITTI state-of-the-art method [36]. With 5% labeled data, our self-supervised model demonstrates over 6.3% and 5.6% relative improvement in mAP compared to PointContrast [44] on Waymo and nuScenes. Our contributions can be summarized as follows:

- We make the first attempt to propose a simple yet effective self-supervised LiDAR-based 3D object detection framework for alleviating the demand for extensive human annotations, towards a more flexible and scalable self-driving system.
- We propose a novel GCC-3D that is the first self-supervised learning mechanism to integrate both geometry-aware structure contrast and harmonized semantic pseudo-instance clustering. This method successfully self-explores and enhances the 3D instance-level representation from both the geometry and semantic perspectives.
- Our GCC-3D framework can achieve state-of-the-art performances on all popular 3D object detection benchmarks, *i.e.*, 67.29% mAP on Waymo (20% labeled data) and 57.3% mAP on nuScenes.

2. Related Work

LiDAR-based 3D Object Detection. This task's objective is to detect objects of interest and localize their amodal 3D bounding boxes from sparse and unorganized point clouds. Some representative works [8, 26, 46] project point clouds to bird's view and use 2D CNNs to learn the point cloud features. Some other works [51, 38] apply 3D

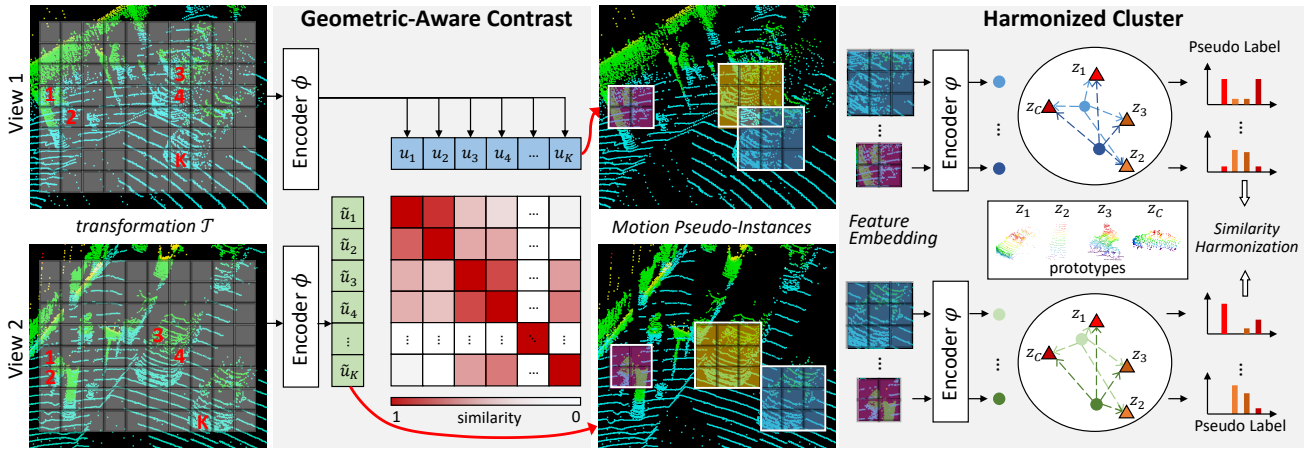

Figure 2. Overview of our GCC-3D self-supervised learning framework. The first key component is the Geometric-Aware Contrast module, where voxels from different views of the same scene are passed through encoder ϕ and we use the geometric distance between them to guide voxel-wise feature learning via a geometric-aware contrastive objective. In the second Harmonized Instance Clustering module, we exploit sequential information to generate pseudo instances in the scenes. The pre-trained voxel features located in each instance will be aggregated as instance embedding and passed through backbone φ for semantic clustering. A harmonization term is introduced to encourage that different views of pseudo-instances should have consistent similarities to clustering prototype centers. These two modules endow our model with both geometric structure and contextual semantic representation.

CNNs over point cloud voxels to generate cuboids. However, these state-of-the-art methods rely on sufficient training labels and precise 3D annotations, which cost a heavy workforce. In this work, we suggest pre-training paradigm is helpful for real-life LiDAR-based 3D object detection and further reduce the pressure of labeling through the proposed self-supervised framework.

Self-supervised Learning. Visual representation learning with self-supervision has drawn massive attention in 2D vision tasks for its fantastic data efficiency and generalization ability. Image-based self-supervised methods design many pretext tasks that exploit their spatial structure [13], color information [11], illumination [15], and rotation [19]. Compared to 2D vision, the limits of big data are far from being explored in 3D. Recent works attempt to adapt the 2D pretext tasks to 3D, but mostly focus on low-level tasks [50, 10, 16] or single object classification tasks [17, 9, 1, 21, 22, 28, 34, 35]. A recent contrastive-learning based method PointContrast [44] demonstrates promising results on a set of indoor scene-level understanding tasks. However, the good performance of [44] depends on partial views setup, which is usually not available in the outdoor autonomous driving scenario. The simple point-level pre-training objective in [44] is not properly designed and can assign points with similar local structure as negative pair, thus raising an obstacle for good contrastive representation learning. And it also neglects semantic information which is important for high-level 3D scene understanding tasks. So in this work, we propose a properly-designed self-supervised learning framework that captures both spatial discriminative information and semantic representation.

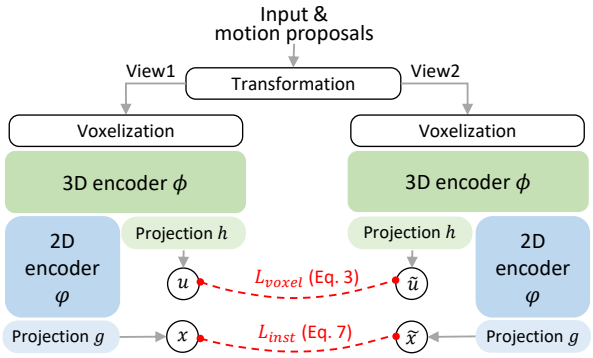

Figure 3. The flowchart of GCC-3D pre-training. We first pre-train 3D encoder ϕ with Geometric-Aware Contrastive objective (eq.3). Then we load the weights to further pre-train 2D encoder φ in Harmonized Instance Cluster module (eq.7).

3. Method

In this section, we elucidate our proposal of the novel self-supervised 3D detection framework GCC-3D that seamlessly integrates the geometry-aware contrast and clustering harmonization, illustrated in Fig. 2. We will first introduce the Geometric-Aware Contrast module that enhances spatial-sensitive local structure representation, then elaborate on the Pseudo-Instance Clustering harmonization mechanism, which encourages different views of pseudo-instances to be consistent.

As with the typical pipeline of LiDAR-based 3D object detection tasks, a 3D encoder ϕ takes the point cloud of a scene as input and estimates 3D bounding box representations for objects, including information on shapes, orientations and semantic classes. The quantized representations

are then reshaped and fed to a 2D backbone φ to produce a feature map $\boldsymbol{F}$. A task-specific head ψ takes $\boldsymbol{F}$ and can either be a 2D anchor-based detector or an anchor-free one.

3.1. Geometric-Aware Contrastive Objective

It is crucial to learn meaningful local structural features of point clouds. Yet methods adopted by previous work [44] focus on hard labeling strategy in contrastive learning, which can be ineffective. This is because, in the hard labeling strategy, both positive and negative pairs can be formed by two voxels extracted from the same type of object, which can be confusing and hinder the network from learning good representations. However, based on the observation that spatially close voxels in the 3D world are more likely to have similar local geometric structures (or belong to the same object), we can use the geometric distance among voxels as a proxy for their feature similarity.

We illustrate our learning process in Fig. 2. Given an original point cloud scene S, we sample a random geometric transformation $\mathcal{T}$ to transform it into an augmented view $\tilde{S}$. We mainly consider similarity transformation including rotation, translation and scaling. We then *voxelize* these two scenes into regular voxels and feed them into a shared 3D encoder ϕ to get voxel-wise features. K voxels are then sampled from the original scenes and we obtain their corresponding voxels in its counterpart $\tilde{S}$ via greedy-searching its nearest voxel center distance. This gives us the corresponding mapping M between two views $(v_i,\tilde{v_i})\in M$, where voxel v_i and voxel $\tilde{v_i}$ are a pair of matched voxels across two views. The voxel-wise features are then projected to a latent space for geometric-aware contrastive learning by a ResMLP [24] h and the final feature for voxel v_i is denoted as $\boldsymbol{u_i}= h(\phi(v_i)) \in \mathbb{R}^{D_1}$.

Then we calculate the Euclidean distance between the center of voxels in the two views, denoted as $d_{i,j} = \parallel \mathcal{T}(v_i) - \tilde{v}_j \parallel$. For each voxel v_i, we softmax the distances between the chosen voxel v_i and all sampled voxels $\tilde{v}_j$ in augmented view to get weights $w_{i,j}(v_i, \tilde{v}_j)$:

$$w_{i,j}(v_i, \tilde{v}_j) = \frac{e^{-d_{i,j}}}{\sum_{(\cdot,\tilde{v}_k)\in M} e^{-d_{i,k}}}. \tag{1}$$

Then the weights $w_{i,j}$ are used to calculate the similarity among voxels by minimizing the soft InfoNCE loss [32]:

$$L(v_i) = -\sum_{(\cdot,\tilde{v}_j)\in M} w_{i,j}(v_i, \tilde{v}_j) log \frac{e^{\boldsymbol{u}_i^\top \tilde{\boldsymbol{u}}_j/\tau}}{\sum\limits_{(\cdot,\tilde{v}_j)\in M} e^{\boldsymbol{u}_i^\top \tilde{\boldsymbol{u}}_j/\tau}}. \tag{2}$$

Since the global transformation brings different offsets between original voxels and their augmented counterparts, instead of taking all matched pairs as equally positive, we use the distances among all the positive pairs to calculate the match confidence $\rho_i=\frac{e^{-d_{i,i}}}{\sum_{(v_j,\tilde{v}_j)\in M} e^{-d_{j,j}}}$ for positive pair $(v_i, \tilde{v}_i)$. The final loss is thus:

$$L_{voxel} = \sum_{(v_i,\cdot)\in M} \rho_i L(v_i). \tag{3}$$

By minimizing L_{voxel}, our 3D encoder ϕ can learn the voxel-wise geometric-aware representation of local structure with equivariance to different transformations.

3.2. Harmonized Pseudo-Instance Clustering

For complex 3D scene understanding task like 3D object detection, simply learning voxel-level geometric feature $\boldsymbol{u}$ might not promise good performance. It is also important to learn contextual semantic information for the model to detect with better robustness. Nevertheless, learning such information requires bounding boxes with exact patches of objects and ground truth semantic label of different objects, both of which unavailable in unsupervised learning settings.

To tackle this problem, we introduce a Motion Pseudo-Instance Generation component into our pipeline. It utilizes sequential information in the dataset to propose pseudo instances. The categorical labels for these instances are obtained by using K-means clustering over instance-level features and we use these labels to further pre-train our model following [41]. However, the hard labeling strategy in [41] takes all the unassigned cluster centers (prototypes) as equally negative. As is discussed previously, it violates the fact that some prototypes can be similar or represent the same semantic classes, especially considering that our number of prototypes C is much larger than the actual semantic class in the scene. Hence, it neglects the heterogeneous similarities between embeddings of pseudo-instances and can lead to the "class collision" [3] problem. Therefore, we propose a Clustering Harmonization mechanism to encourage that different views of pseudo-instances should have consistent feature similarities to clustering prototype centers. As is shown in Fig. 2, we keep using a multi-view setup to learn pseudo-instance representation that is equivariant to transformation and robust to noise.

Motion Pseudo-Instance Generation. In the self-driving environment, the ego-vehicle's sensor status is available at every frame (50 fps). The intuition behind our design is to use sequential information to localize patches with moving objects. Note that moving objects can be recognized from stationary objects by observing the non-overlapping area among consecutive frames. Therefore, we can analyze the occupancy voxel point information in BEV view to identify possible moving voxels and then find out connected domains between neighboring moving voxels to obtain pseudo moving patches.

Specifically, given two consecutive LiDAR frame p and q, the rigid transformation between their coordinates can be

initial.	Model	0.05		0.1		0.2		0.5		1	
		AP/L2	APH/L2	AP/L2	APH/L2	AP/L2	APH/L2	AP/L2	APH/L2	AP/L2	APH/L2
random init.	cppp [49]	49.30	44.35	55.66	51.14	59.14	55.25	61.00	56.94	62.79	58.91
PointContrast [44]		50.10	44.97	56.82	52.35	60.04	56.31	61.83	57.16	63.10	58.97
GCC-3D		52.92$^{+3.72}$	47.85$^{+3.50}$	58.68$^{+3.02}$	53.89$^{+2.75}$	61.58$^{+2.44}$	57.39$^{+2.14}$	63.66$^{+2.66}$	59.73$^{+2.79}$	64.17$^{+1.38}$	60.46$^{+1.55}$
random init.	cpvoxel [49]	43.13	40.27	50.51	47.73	58.90	56.28	63.60	61.09	66.29	63.80
GCC-3D		44.70$^{+1.57}$	41.75$^{+1.48}$	54.09$^{+3.50}$	51.34$^{+3.61}$	60.86$^{+1.94}$	58.19$^{+1.91}$	65.45$^{+3.61}$	62.85$^{+1.76}$	67.00$^{+3.61}$	64.54$^{+0.76}$

Table 1. Main results of 3D detection on Waymo val set. "cppp" and "cpvoxel" indicate Centerpoint with Pointpillars and VoxelNet. We train 36 epochs for cppp and 12 epochs for cpvoxel.

written as: $T = T_{(lidar_p \leftarrow ego_p)} T_{(ego_p \leftarrow ego_q)} T_{(ego_q \leftarrow lidar_q)}$, where we align the frame q into the coordinate system of p by $p' = T(q)$. We quantify LiDAR points p and p' into regular voxels and compute the average coordination of points in each voxel. Voxels with an average coordination distance between p and p' larger than a predefined threshold will be considered as moving voxels. Finally, we group these voxels as pseudo instances by eight-neighbor-hooding [4].

Clustering Harmonization. After passing the point cloud scene S through 3D encoder ϕ and 2D backbone φ, we obtain its feature map $\boldsymbol{F}$. Given the set of pseudo-instance positions P in scene S obtained in the instance generation module, we obtain instance-level embedding $\boldsymbol{x}$ by cropping features corresponding to each pseudo instance on feature map $\boldsymbol{F}$. These embeddings are RoIAligned and projected to a latent embedding space by an MLP g to obtain instance-level feature $\boldsymbol{x} = g(RoIAlign(\boldsymbol{F}, P^m)) \in \mathbb{R}^{D_2}$ for pseudo instance m. These instance-level features are clustered into C distinct groups based on a geometric criterion at the end of each epoch. A $D_2 \times C$ prototype matrix Z and the cluster assignments y for each instance are then obtained. These assignments y will be used as pseudo-labels for training the pseudo-instance clustering network. With the multi-view setup, we obtain augmented views of instance-level feature $\tilde{\boldsymbol{x}}$ by passing augmented scene $\tilde{S}$ through our network using transformed instance position $\tilde{P}$, then crop the features on the corresponding feature map $\tilde{F}$.

To capture similarities among embeddings of pseudo-instances, we propose a harmonization term that encourages different views of pseudo-instances to be consistent with their clustering prototype centers. Specifically, given a pseudo instance m and the prototype matrix $Z \in \mathbb{R}^{D_2 \times C}$, we calculate the similarity between instance feature $\boldsymbol{x}$ and the prototypes $z_i (i \in \{1, ..., C\})$ as:

$$J(i) = \frac{e^{\boldsymbol{x}^\top \boldsymbol{z}_i}}{\sum_{k=1}^{C} e^{\boldsymbol{x}^\top \boldsymbol{z}_k}}, \tag{4}$$

where $J(i)$ is the probability that embedding $\boldsymbol{x}$ is assigned to cluster center i. Similarly, the probability of assigning the augmented view of instance feature $\tilde{\boldsymbol{x}}$ to this cluster is:

$$H(i) = \frac{e^{\tilde{\boldsymbol{x}}^\top \boldsymbol{z}_i}}{\sum_{k=1}^{C} e^{\tilde{\boldsymbol{x}}^\top \boldsymbol{z}_k}}. \tag{5}$$

initial.	Model	0.05		0.1		0.5		1	
		mAP	NDS	mAP	NDS	mAP	NDS	mAP	NDS
random init.	cppp [49]	25.79	34.35	37.12	49.14	46.29	57.25	49.61	60.20
PointContrast [44]		30.79	41.57	38.25	50.10	47.94	58.24	50.09	60.33
GCC-3D		32.75	44.20	39.14	50.48	48.48	58.87	50.84	60.76
random init.	cpvoxel [49]	38.01	44.34	46.85	55.51	54.78	62.92	56.19	64.48
PointContrast [44]		39.75	45.05	47.74	55.98	54.97	63.53	56.25	64.40
GCC-3D		41.10	46.81	48.43	56.71	55.87	64.50	57.26	65.01

Table 2. Data-efficient 3D detection on nuScenes val set. We show the NDS, mAP for all classes. "cppp" and "cpvoxel" indicate Centerpoint with Pointpillars and VoxelNet.

We further introduce a harmonization term that encourages consistency between the assignment probability J and H via symmetric Kullback–Leibler divergence:

$$L_{harmo}(x) = \frac{1}{2} D_{KL}(J \parallel H) + \frac{1}{2} D_{KL}(H \parallel J). \tag{6}$$

This term not only encourages networks to learn features with equivariance to a set of transformations, but also considers the similarity between different prototype centers, thus alleviating "class collision" [3] problem. It fits well into the current self-clustering framework and our final loss is a weighted average of clustering loss term and the consistency regularization term:

$$L_{inst} = \sum_{m \in P} l(\boldsymbol{x}, y) + l(\tilde{\boldsymbol{x}}, y) + \alpha L_{harmo}(\boldsymbol{x}), \tag{7}$$

where l is the cross entropy loss and y is the cluster assignment of instance feature $\boldsymbol{x}$.

Combination with the Geometric-Aware Contrastive Objective. Fig. 3 presents the flow of the pre-train process. Before pseudo-instance clustering pretraining, we first load weights of 3D encoder ϕ that gets pre-trained on the geometric-aware contrast module and provides discriminative voxel-level structure feature. We then use the harmonized pseudo-instance clustering objective to further pre-train both the 3D encoder ϕ and the 2D backbone φ. The weights will be used as initialization for the finetune stage.

4. Experiments

Pre-training Details. In the Geometric-Aware Contrasting module, $K = 1024$ and $D_1 = 64$. We pre-train the model for 20 epochs and use Adam optimizer with the initial learning rate 0.001. The batch size is 6 and τ is 1. In

Model	mAP	Car AP/APH	Pedestrian AP/APH	Cyclist AP/APH
SECOND [45]	55.08	59.57/59.04	53.00/43.56	52.67/51.37
PART^2 [37]	60.39	64.33/63.82	54.24/47.11	62.61/61.35
PV-RCNN [36]	59.84	64.99/64.38	53.80/45.14	60.72/59.18
centerpoint-voxel [49]	63.46	61.81/61.30	63.62/57.79	64.96/63.77
centerpoint-voxel 2stage [49]	64.63	64.70/64.11	63.26/58.46	65.93/64.85
GCC-3D (PV-RCNN)	**61.30**$^{+1.46}$	**65.65/65.10**	**55.54/48.02**	**62.72/61.43**
GCC-3D (centerpoint-voxel)	**65.29**$^{+1.83}$	**63.97/63.47**	**64.23/58.47**	**67.68/66.44**
GCC-3D (centerpoint-voxel 2stage)	**67.29**$^{+2.66}$	**66.45/65.93**	**66.82/61.47**	**68.61/67.46**

Model	All mAP	All NDS
WYSIWYG [25]	35.0	41.9
3DSSD [48]	42.6	56.4
HotSpotNet [7]	50.6	59.8
CBGS [52]	50.6	62.3
centerpoint-pp [49]	49.6	60.2
centerpoint-voxel [49]	56.2	64.5
GCC-3D (centerpoint-pp)	**50.8**$^{+1.2}$	**60.8**$^{+0.6}$
GCC-3D (centerpoint-voxel)	**57.3**$^{+1.1}$	**65.0**$^{+0.5}$

Table 3. Comparison with 3D detection on **20%** Waymo (Left) and 100% nuScenes (Right). All methods train 30 epochs following PCDet and 20 epochs for nuScenes. "pp" indicates Pointpillar and "voxel" means VoxelNet using as encoder ϕ.

Harmonized Pseudo-Instance Clustering module, we pre-train for 20 epochs with Adam optimizer. The initial learning rate is 0.0048 with a cosine decay. The prototype number C is 100, D_2 is 128 and α is 0.1. All experiments run on 8 NVIDIA V100 GPUs. We use VoxelNet and PointPillars in CenterPoint network [49] as backbones, denoted as CenterPoint-pp and CenterPoint-voxel separately. We conduct experiments on two most popular self-driving datasets: Waymo Open Dataset [39] and nuScenes Dataset [5].

4.1. Data-Efficient 3D Object Detection Benchmark

To formally explore data-efficient scene understanding in autonomous driving, we propose a 3D object detection benchmark with limited bounding box annotations. Specifically, for each dataset, only a limited fraction of scenes will be labeled and we consider the configurations including $\{0.05, 0.1, 0.2, 0.5\}$ (1 represents the entire train set). We pre-train our model as initialization for fine-tune and compare against the baseline of train from scratch. The training schedules and setup during the fine-tuning stage follow [49]. During test time, evaluation is performed on all scenes in the validation set. Table 1 and 2 summarize our results.

On the Waymo validation set, our model brings consistent improvement over the baseline model with both PointPillars and VoxelNet encoders. Specifically, with 50% of labels, the PointPillars model pre-trained with our method achieves 63.66% mAP, outperforming baseline with 100% labels. The performance gap does not diminish when more box annotations are available. Similar behaviors can be observed on the nuScenes dataset, and the difference between with and without our pre-train is more pronounced. As shown in Table 2, the trained-from-scratch detector can barely produce any meaningful results when the data is scarce (*e.g.*, 5% or 10%). However, fine-tuning with our pre-trained weights, PointPillars can perform significantly better (*e.g.*, improve mAP by 6.96 % with 5% labeled data).

4.2. Comparison with SOTA

We compare our method with other state-of-the-art models on LiDAR-based 3D Object Detection in Table 3. For Waymo, we follow the training schedule in PCDet[1]. Experiments on nuScenes are implemented in CenterPoint[2].

pretrain / ft.	KITTI (mode mAP)	nuScenes (NDS)	Waymo (mAPH/L2)
random init.	69.77	45.55	63.80
nuScens→	70.75+0.98	45.69+0.14	64.32+0.52
Waymo→	71.26+1.49	45.65+0.10	64.54+0.64

Table 4. Transfer pre-trained weights from dataset A (column) to the whole set B (row). We use CenterPoint-voxel for nuScenes and Waymo, and PV-RCNN for KITTI. We show results are evaluated on moderate difficulty mAP for KIITI, NDS for nuScenes and mAPH under L2 difficulty case for Waymo.

After GCC-3D pre-trained on Waymo, several state-of-the-art 3D object detectors demonstrate better performance over their training from scratch baselines (+1.83% mAP on one-stage CenterPoint, +2.66% mAP on two-stage CenterPoint, +1.46%mAP on PV-RCNN [36]), showing strong generalization ability of our pre-train method. The performance of our training from scratch baseline based on CenterPoint is already higher than SECOND [45], PART^2 [37] and PV-RCNN. After initialization with GCC-3D pre-train, one-stage CenterPoint reaches 65.29% mAP and 62.79% mAPH (with +1.83% mAP and +1.84% mAPH improvement over the baseline) and is even better than two-stage CenterPoint baseline model. The two-stage CenterPoint module pre-trained with our method reaches 67.29% mAP and 64.95% mAPH. A similar phenomenon is shown on nuScenes. The CenterPoint [49], first place on nuScenes 3D object detection benchmark, initialized with our GCC-3D pre-train achieves a higher performance of 57.3% mAP.

4.3. Transfer Datasets and Models

We evaluate our representation on different datasets to assess whether the features learned on the source domain are generic and thus applicable across the target domain. We use CenterPoint-voxel [49] during pre-train and load weights of 3D encoder ϕ and 2D backbone φ as initialization for finetune. When finetuned on KITTI [18], we use

[1] https://github.com/open-mmlab/OpenPCDet
[2] https://github.com/tianweiy/CenterPoint

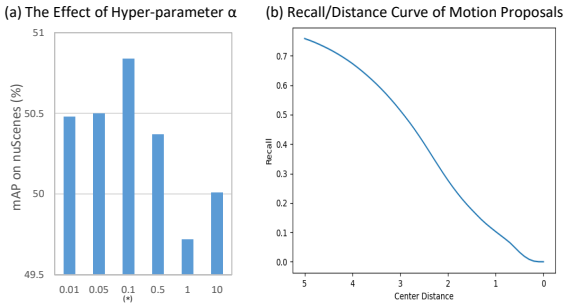

Figure 4. (a) Ablation studies on hyper-parameter α for our GCC-3D method. (*) indicates default value. (b) The recall and center distance curve between generated instance and groundturth.

task-specific head of PV-RCNN [36] and follow training setups in PCDet [40]. When finetuned on nuScenes, each scene only uses one sweep of point cloud scene. More experiment setups can be found in the appendix.

Table 4 shows our GCC-3D reaches consistent improvement cross different datasets than random initialization. We observe that 1) Leveraging pre-trained weights on large scale dataset then finetune on small datasets can bring more significant performance improvements: when pretrained on nuScenes and Waymo then finetuned on KITTI, we see +0.98% mAP and +1.49% mAP improvement respectively; 2) Model pre-trained on nuScenes and fine-tuned on itself show bigger improvement over baseline than model pre-trained on Waymo (+0.14% *vs.* +0.10%). We conjecture this dilution of pretraining effectiveness arises from domain gaps between different point clouds (point clouds of Waymo are much denser than that of nuScenes).

4.4. Ablation Study

In this section, we conduct ablation study experiments to analyze the effectiveness of different modules and hyperparameters of GCC-3D.

Effect of Geometry-Aware Contrast and Harmonized Pseudo-instance Clustering. We pre-train separately with Geometry-aware Contrast module and Harmonized pseudo-instance Clustering module following the same pre-train setup as in GCC-3D. Then we fine-tune the models on Waymo and nuScenes datasets with limited annotations (5%) and evaluate on the full validation set. Results are in Table 5. While the Harmonized Pseudo-instance Clustering module achieves reasonable pre-train performance, the Geometry-Aware Contrast module boosts the fine-tune result more significantly, with +6.77% mAP and +2.59% mAP than baseline on nuScenes and Waymo separately. Meanwhile, combining these two modules improves the performance to 32.75% mAP on nuScenes and 52.92% mAP on Waymo. It demonstrates that our method of integrating geometry structure and semantic context representation helps the high-level 3D Object detection task.

Method	Waymo		nuScenes	
	mAP	mAPH	mAP	NDS
random init.	49.30	44.35	25.79	34.35
PointContrast [44]	50.10	44.97	30.79	41.57
DeepCluster [41]	49.26	44.31	27.84	38.19
SwAV [6]	-	-	27.41	35.60
Geometry-Aware	$50.32^{+1.02}$	$45.21^{+0.86}$	$32.56^{+6.77}$	$43.81^{+9.46}$
Harmonization Term	$51.89^{+2.59}$	$46.82^{+2.47}$	$30.32^{+5.77}$	$42.07^{+7.72}$
GCC-3D	$\mathbf{52.92}^{+3.62}$	$\mathbf{47.85}^{+3.50}$	$\mathbf{32.75}^{+6.96}$	$\mathbf{44.2}^{+9.85}$

Table 5. Ablation study of different modules and comparison with other self-supervised learning method on 5% nuScenes and Waymo annotations. All results are based on Centerpoint-pp.

Effectiveness of Hyper-parameters. We study the effect of hyper-parameter introduced in GCC-3D: the coefficient of harmonization term α. In Fig. 4, we show the effect of α on nuScenes with 100% annotation. It is finetuned with the CenterPoint-pp model for 20 epochs. The best coefficient is 0.1. We see that by using the harmonization term, the objective accuracy can be boosted from 50.48% mAP to 50.84% mAP. Increasing α to 0.5 and beyond can hurt the performance. We conjecture that this is because when $\alpha \geq 0.5$, the model is over-regularized by the harmonization term and loses some discrimination among categories.

4.5. Exploration on Click-supervised Pretraining

To alleviate the annotation burden in 3D object detection, some efforts click the object center to provide location supervision for this task [42, 27, 31]. Inspired by these works, we propose a simple supervised pretraining baseline pre-trained with centers of 3D objects of full dataset and fine-tuned on a limited scale of fully-annotated data.

We compare the performance of our GCC-3D with the Click-supervision pre-train on 3D detection task based on CenterPoint-pp. The results are in Fig. 1. Although the Click-supervision method achieves good performance with a highly limited fraction of data, our GCC-3D method show better performance when we increase the finetuned data fraction. We believe it is because our method can learn more robust features than click-annotation pre-train. The latter enforces the network to focus on localization regression task, but our method can learn more robust embedding suitable for 3D object detection task, which requires both localization, classification and rotation representation.

4.6. Comparison with Other SSL Methods

GCC-3D is the first self-supervised pre-training framework tailored for LiDAR-based 3D Object Detection. Nevertheless, we re-implement and adopt previously published self-supervised learning models, including contrastive method (PointContrast [44]) and clustering-based methods (Deepclusterv2 [41] and SwAV [6]) on 3D object detection tasks. These methods are actually closely related

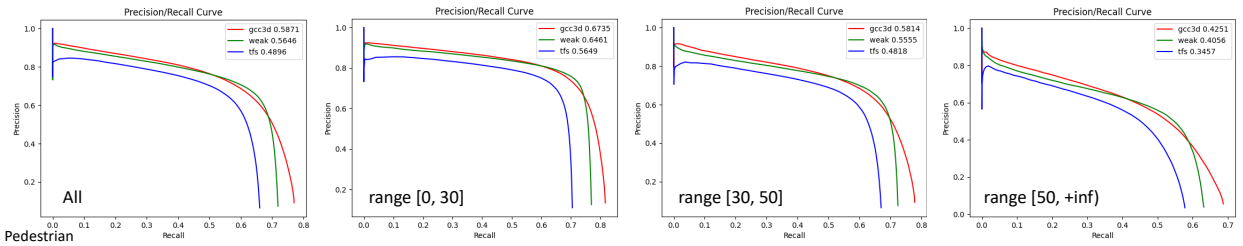

Figure 5. Precision and Recall Curve of pedestrian class on 20% Waymo val set.

to our framework. The PointContrast method can be implemented by revising the voxel-level contrastive learning in our Geometry-aware Contrast module with hard labeling. During pre-train, it is optimized following the same setup as in GCC-3D. For Deepcluster v2 and SwAV, instead of applying these cluster-based learning strategies on images, we use them on our proposed pseudo-instances. In Harmonized pseudo-instance Clustering module, getting rid of the harmonization term and choosing α to be 0 reduce to Deepclusterv2. SwAV is an adaptation of Deepclusterv2 with online clustering. During pre-train, their hyperparameters and optimization setup follow the same setting as in GCC-3D.

We compare with these self-supervised pretraining methods on Waymo and nuScenes dataset with finetuning on limited annotation (5%) based on CenterPoint-pp in Table 5. Our pre-training method outperforms all these pre-training strategies, achieving 32.75% mAP on nuScenes and 52.92% mAP on Waymo with only 5% annotations. Notably, the consistent improvement of Geometry-Aware Contrast over PointContrast (32.56% mAP *vs.* 30.79% mAP on nuScenes and 51.89% mAP *vs.* 50.10% mAP on Waymo) demonstrates the effectiveness of our geometry-aware design – using distance to guide feature similarity can alleviate inconsistent hard labeling problem in PointContrast. Harmonization Clustering produces better performance than Deepclusterv2 and SwAV, which proves the importance of our proposed harmonization term.

4.7. How Pre-training Affects 3D Object Detection

Speed Up Convergence with Better Performance. To investigate whether the advantage of pre-training recedes when we prolong the fine-tuning stage, we compare the train from scratch baseline with fine-tuned model initialized with Click-supervised pre-training in Fig. 6. The experiments are conducted on nuScenes dataset with 2.5% and 5% labeled data during fine-tune and 20 times bigger scale dataset during pre-train. We train until both models converge and find that the model with pre-training consistently outperforms the baseline. This observation is not consistent with [23], which claims that train-from-scratch baselines are no worse than their pre-trained counterparts. Moreover, it is observed that pretraining speeds up convergence by four times compared with baseline (20 epochs *vs.* 80 epochs). These observations motivate using pre-training to

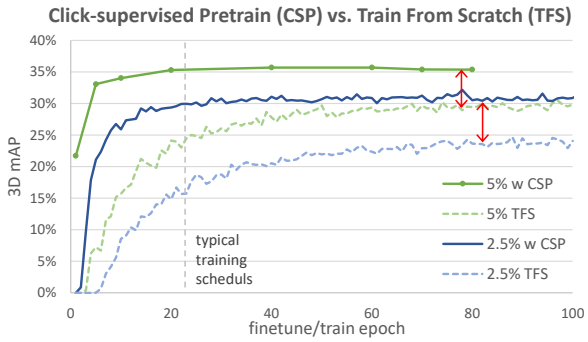

Figure 6. Fine-tuned on 2.5% and 5% nuScenes dataset.

reduce resource consumption for real-world application.

Consistent Improvement with Large Finetune Scale by Unsupervised Pre-train. It is observed in some pretrain methods that as the fine-tuned data scale grows, the benefits of pre-training will decrease. We see a similar phenomenon on Click-supervised pre-train. However, in our 3D detection baseline, we can observe a consistent improvement of the pre-trained model over the baseline. We hypothesize that our unsupervised pre-train objective can learn more robust features and is less likely to overfit to specific tasks.

Alleviate False Positive. To compare the performance of different initializations, including train-from-scratch (tfs), supervised pretraining, and GCC-3D pretraining, we consider the Precision-Recall (PR) curve on Waymo dataset, which plots precision against the recall at different thresholds. Fig. 5 shows that pre-trained models outperform tfs at the same recall level, especially on far range detection. Tfs fails on recall > 0.7 while GCC-3D can help the model make a more precise prediction.

5. Conclusion

In this work, we focus on data-efficient LiDAR-based 3D object detection through a novel self-supervised framework that integrates Geometry-Aware Contrast and Harmonized Pseudo-Instance Clustering. It can capture spatial-sensitive representation and high-level context information. We show the effectiveness of pre-train and hope these findings can drive more research on unsupervised 3D representation learning and 3D scene understanding in the future.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 1, 2

[2] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4253–4262, 2020. 1

[3] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019. 1, 3.2, 3.2

[4] Gregory A Baxes. *Digital image processing: principles and applications*. John Wiley & Sons, Inc., 1994. 3.2

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020. 1, 4

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33, 2020. 4.4, 4.6

[7] Qi Chen, Lin Sun, Zhixin Wang, Kui Jia, and Alan Yuille. object as hotspots. In *ECCV*, 2020. 4

[8] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 2

[9] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 1, 2

[10] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 602–618, 2018. 1, 2

[11] Aditya Deshpande, Jason Rock, and David Forsyth. Learning large-scale automatic image colorization. In *ICCV*, pages 567–575, 2015. 2

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[13] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015. 2

[14] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655, 2014. 1

[15] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *TPAMI*, 38(9):1734–1747, 2016. 2

[16] Gil Elbaz, Tamar Avraham, and Anath Fischer. 3d point cloud registration for localization using a deep neural network auto-encoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640, 2017. 1, 2

[17] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3d point cloud processing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118, 2018. 1, 2

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 1, 4.3

[19] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 2

[20] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 1

[21] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018. 1, 2

[22] Kaveh Hassani and Mike Haley. Unsupervised multi-task feature learning on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8160–8171, 2019. 2

[23] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *ICCV*, pages 4918–4927, 2019. 4.7

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3.1

[25] Peiyun Hu, Jason Ziglar, David Held, and Deva Ramanan. What you see is what you get: Exploiting visibility for 3d object detection. In *CVPR*, pages 11001–11009, 2020. 4

[26] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, pages 12697–12705, 2019. 2

[27] J. Lee, S. Walsh, A. Harakeh, and S. L. Waslander. Leveraging pre-trained 3d object detection models for fast ground truth generation. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018. 4.5

[28] Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9397–9406, 2018. 2

[29] Xia Li, Yibo Yang, Qijie Zhao, Tiancheng Shen, Zhouchen Lin, and Hong Liu. Spatial pyramid based graph reasoning for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8950–8959, 2020. 1

[30] D Matthew and R Fergus. Visualizing and understanding convolutional neural networks. In *ECCV*, pages 6–12, 2014. 1

[31] Qinghao Meng, Wenguan Wang, Tianfei Zhou, Jianbing Shen, Luc Van Gool, and Dengxin Dai. Weakly supervised 3d object detection from lidar point cloud. In *ECCV*, 2020. 1, 4.5

[32] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3.1

[33] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12674–12684, 2020. 1

[34] Aditya Sanghi. Info3d: Representation learning on 3d objects using mutual information maximization and contrastive learning. In *European Conference on Computer Vision*, pages 626–642. Springer, 2020. 2

[35] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. *arXiv preprint arXiv:1901.08396*, 2019. 2

[36] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 4, 4.2, 4.3

[37] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 4, 4.2

[38] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015. 2

[39] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 1, 4

[40] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 4.3

[41] Kai Tian, Shuigeng Zhou, and Jihong Guan. Deepcluster: A general clustering framework based on deep learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 809–825. Springer, 2017. 1, 3.2, 4.4, 4.6

[42] Bernie Wang, Virginia Wu, Bichen Wu, and Kurt Keutzer. Latte: Accelerating lidar point cloud annotation via sensor fusion, one-click annotation, and tracking. *arXiv preprint arXiv:1904.09085*, 2019. 4.5

[43] Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, Haibin Ling, and Ruigang Yang. Salient object detection in the deep learning era: An in-depth survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1

[44] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas J Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. *arXiv preprint arXiv:2007.10985*, 2020. 1, 1, 2, 3.1, 3.2, 3.2, 4.4, 4.6

[45] Y. Yan, Y. Mao, and B. Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10), 2018. 4, 4.2

[46] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. 2

[47] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095, 2020. 1

[48] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 4

[49] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *arXiv:2006.11275*, 2020. 1, 3.2, 3.2, 4, 4, 4.1, 4.2, 4.3

[50] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1802–1811, 2017. 1, 2

[51] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, pages 4490–4499, 2018. 2

[52] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019. 4

[53] Ye Zhu, Sven Ewan Shepstone, Pablo Martínez-Nuevo, Miklas Strøm Kristoffersen, Fabien Moutarde, and Zhuang Fu. Multiview based 3d scene understanding on partial point sets. *arXiv preprint arXiv:1812.01712*, 2018. 1, 1